

SPQR: горизонтальное масштабирование PostgreSQL

Денис Волков,
Open Source RDBMS
Development Team Yandex
Cloud

Кирилл Решке,
Open Source RDBMS
Development Team Yandex
Cloud



Яндекс



1. Дисклеймер
2. Что мы хотим от шардирования?
3. Что мы пробовали?
4. Stateless Postgres Query Router
5. Демо
6. Открытый процесс разработки

Open source RDBMS development at Yandex Cloud

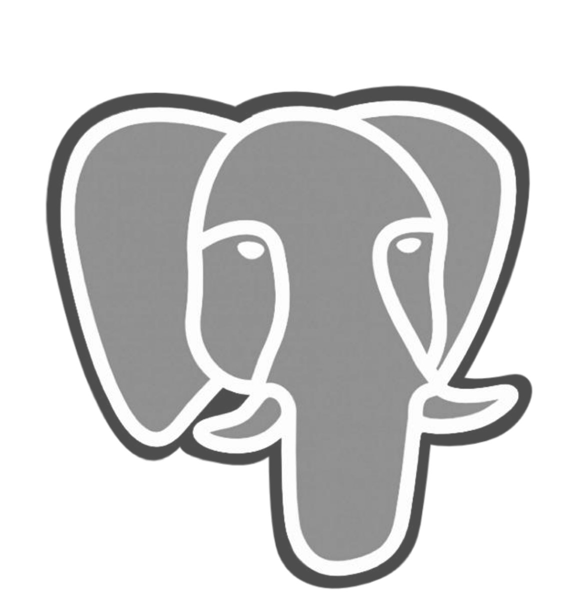
- PostgreSQL, Greenplum contributors
- WAL-G, Odyssey maintainers



PostgreSQL at Yandex Cloud

- Миллионы QPS
- Петабайты данных
- Тысячи кластеров

  От 0,5 до 80 CPU-ядер
- В основном HA-кластеры

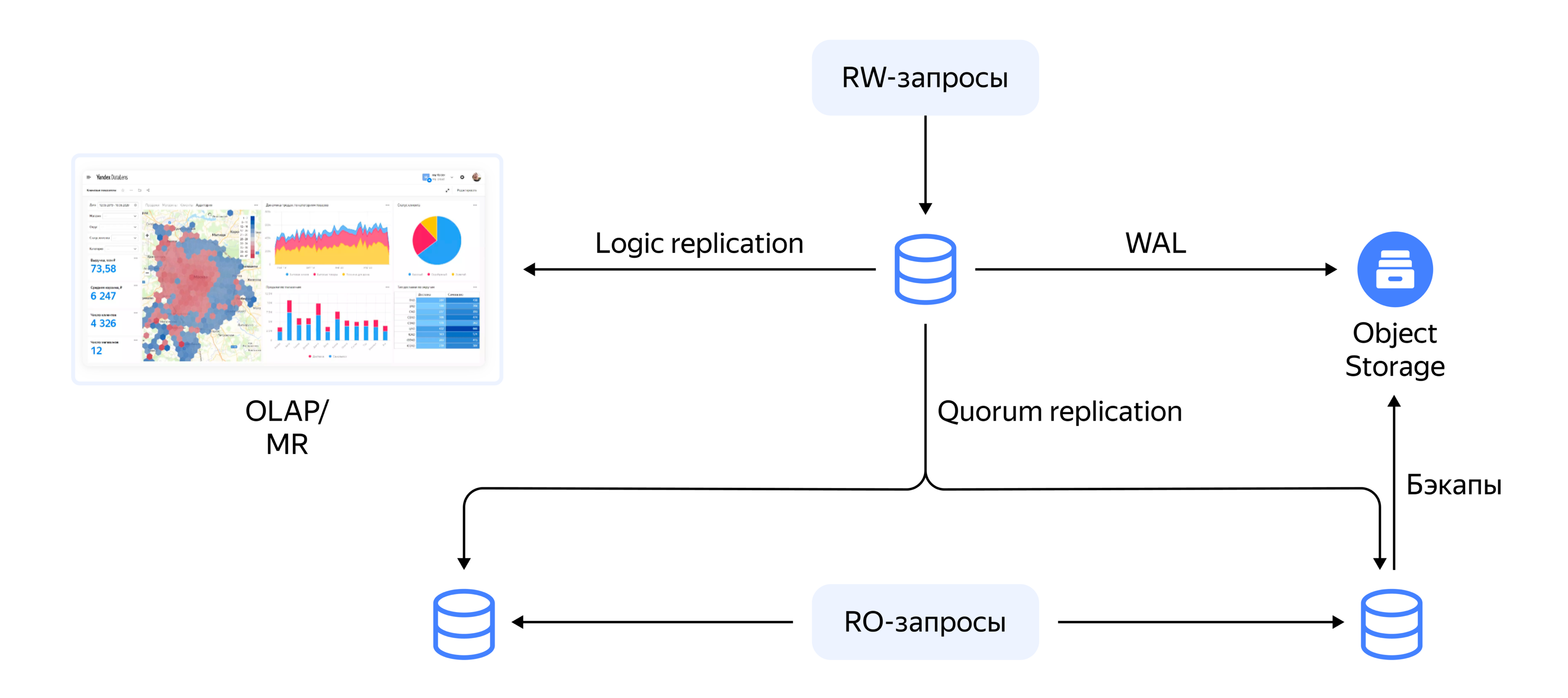


Зачем?

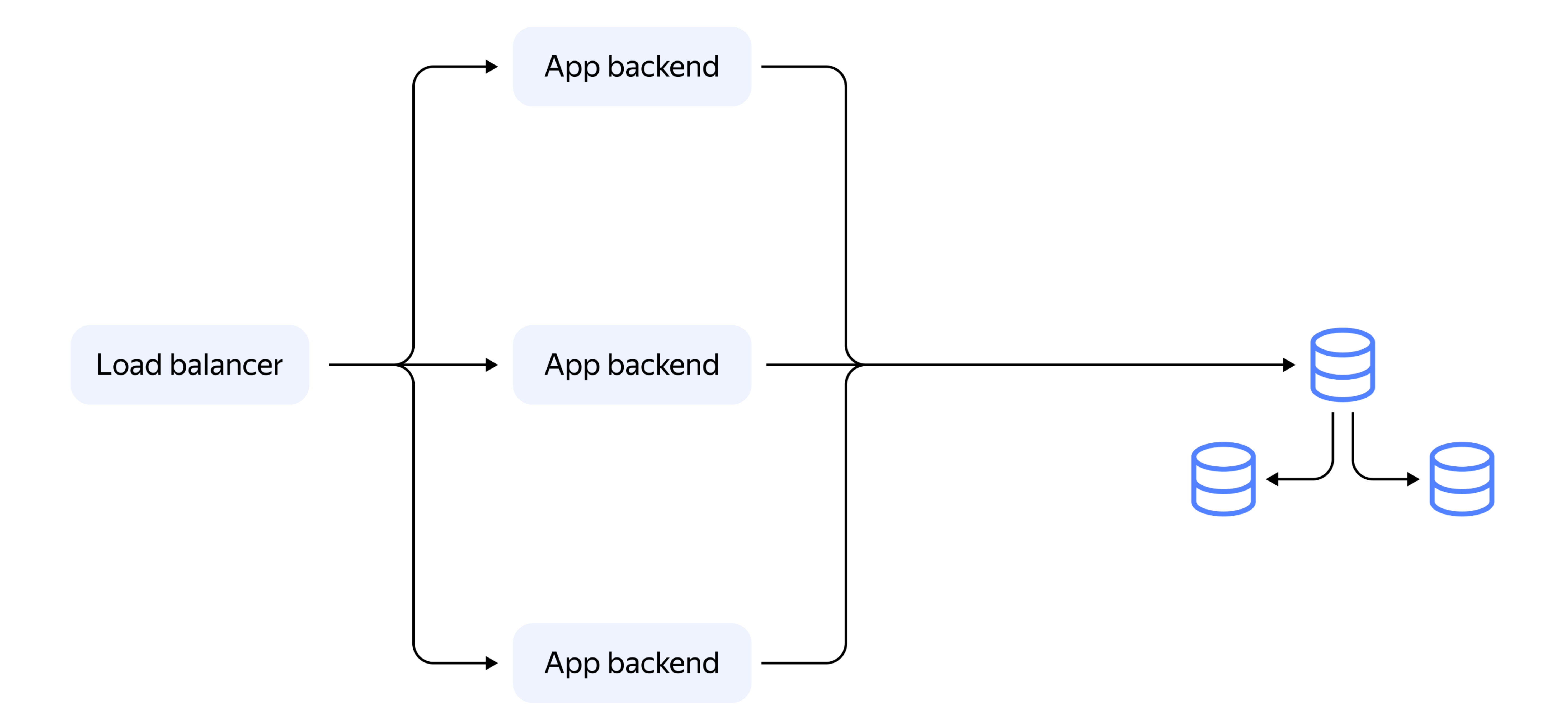
Очень надо

1. Дисклеймер
2. Что мы хотим от шардирования?
3. Что мы пробовали?
4. Stateless Postgres Query Router
5. Демо
6. Открытый процесс разработки

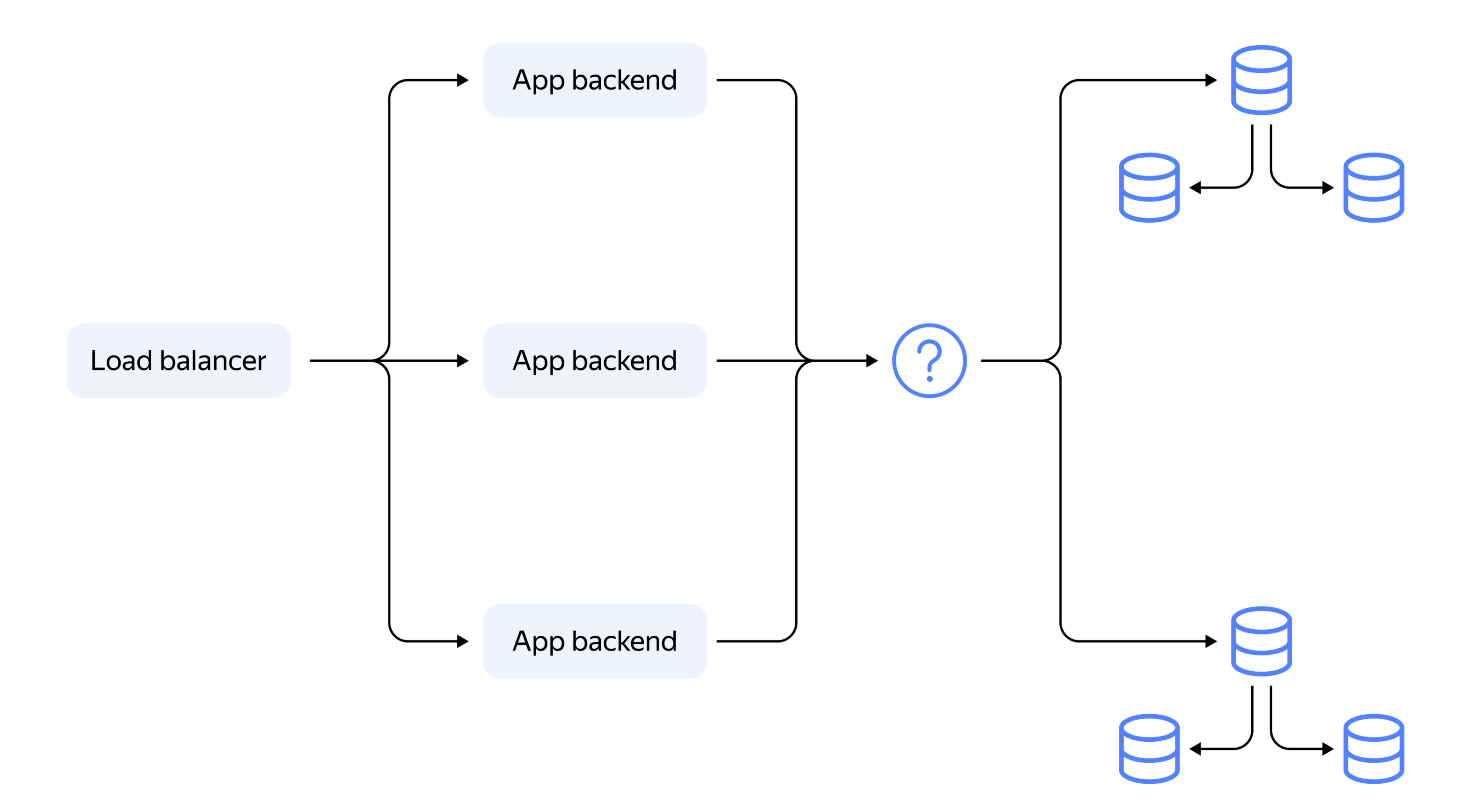
Типичный кластер PostgreSQL



Типичное приложение

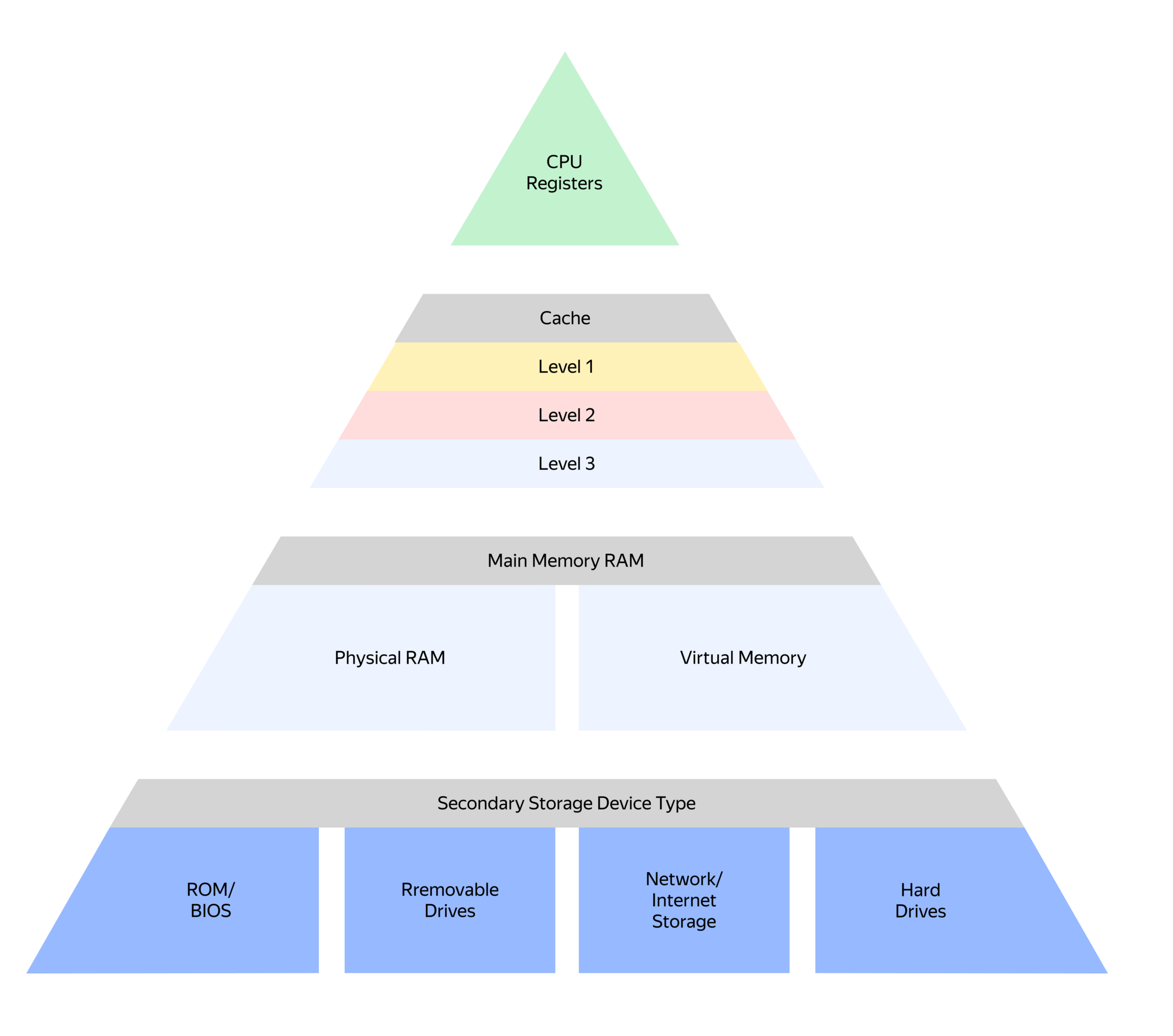


Типичное приложение



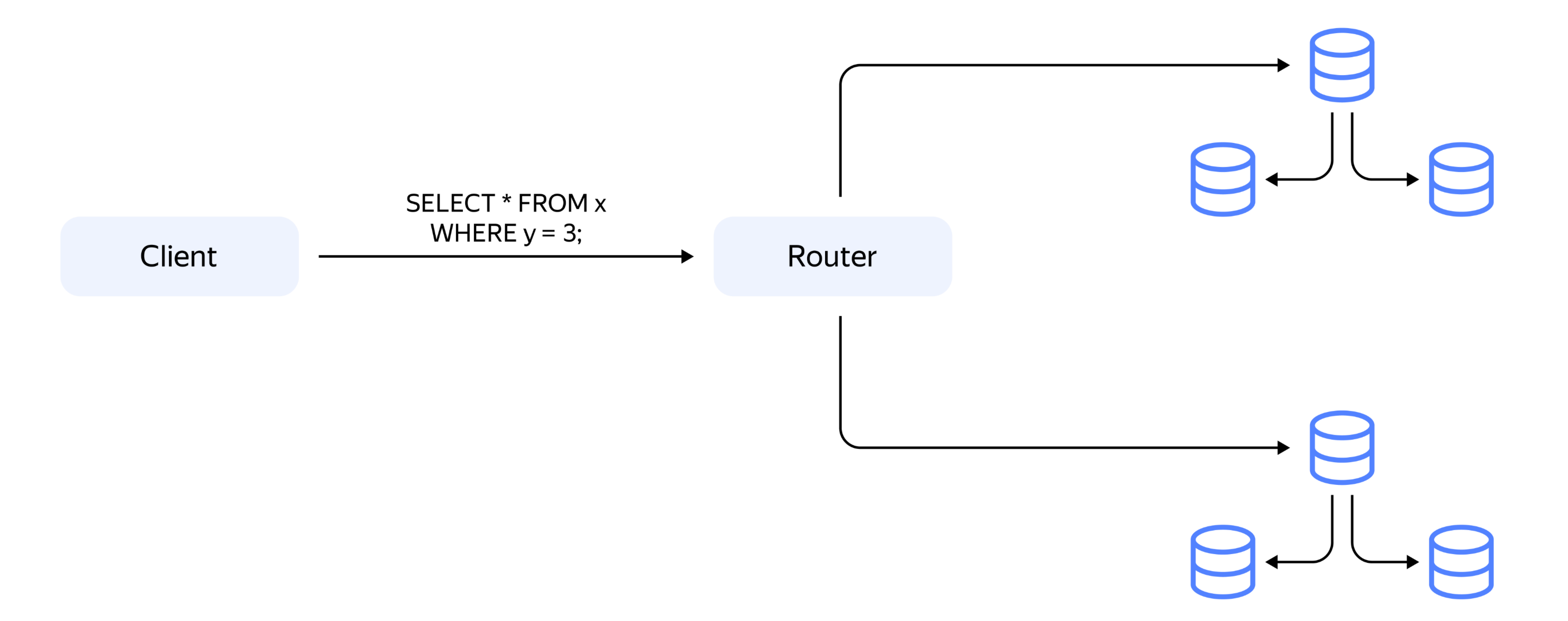
У разных шардов разные ресурсы

- Computational power
- RAM
- Disk IO
- Network throughput



1. Дисклеймер
2. Что мы хотим от шардирования?
3. Что мы пробовали?
4. Stateless Postgres Query Router
5. Демо
6. Открытый процесс разработки

Basic topology



FDW based approach

```
INSERT INTO t VALUES (1, now()):
INSERT data RETURNING ctid;
UPDATE WHERE ctid = …;
```

Planner hook based solution

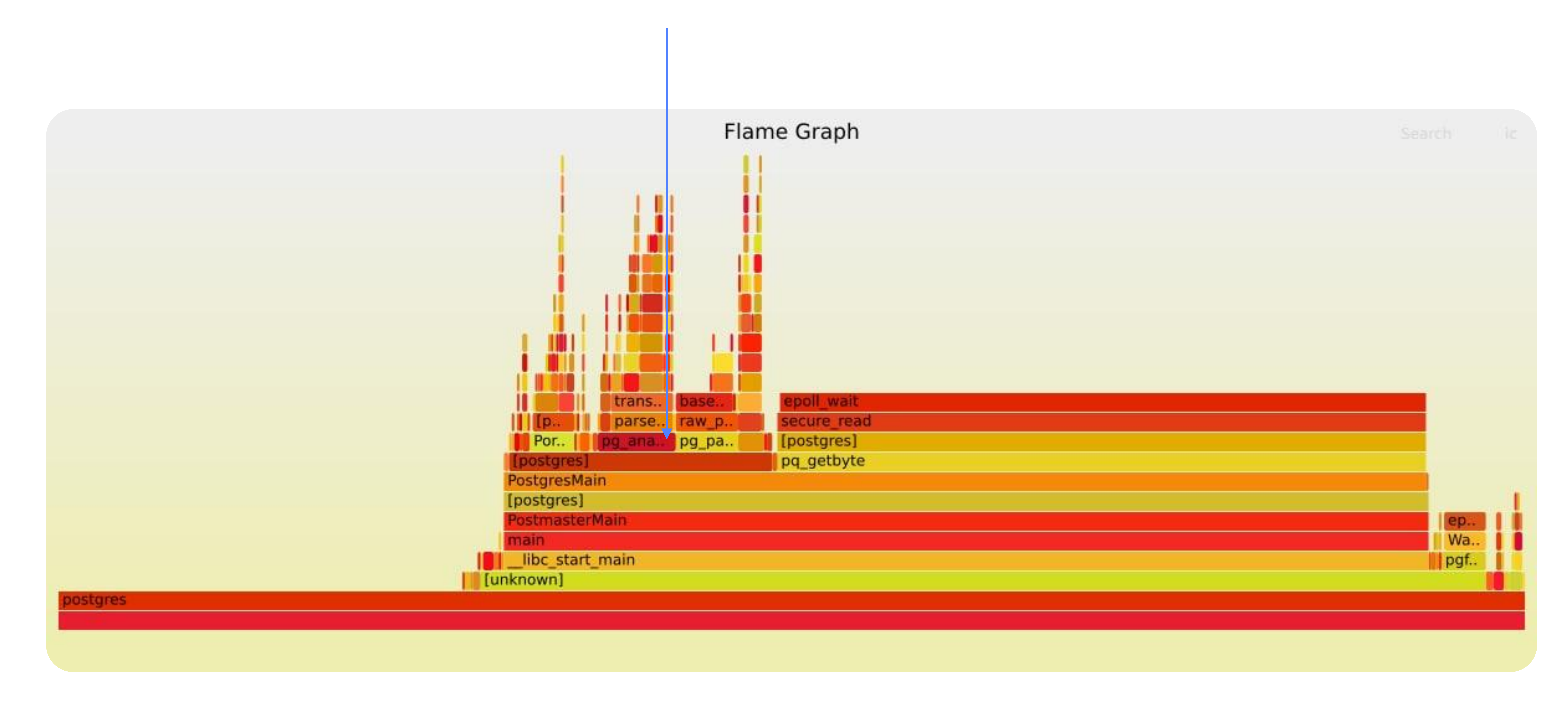


- PostgreSQL Extension
- Custom Exec Nodes
- 5000–6000 TPC on 40 000 Warehouse
- Pain with plain C



github.com/reshke/router

pg_analyze_and_rewrite ()

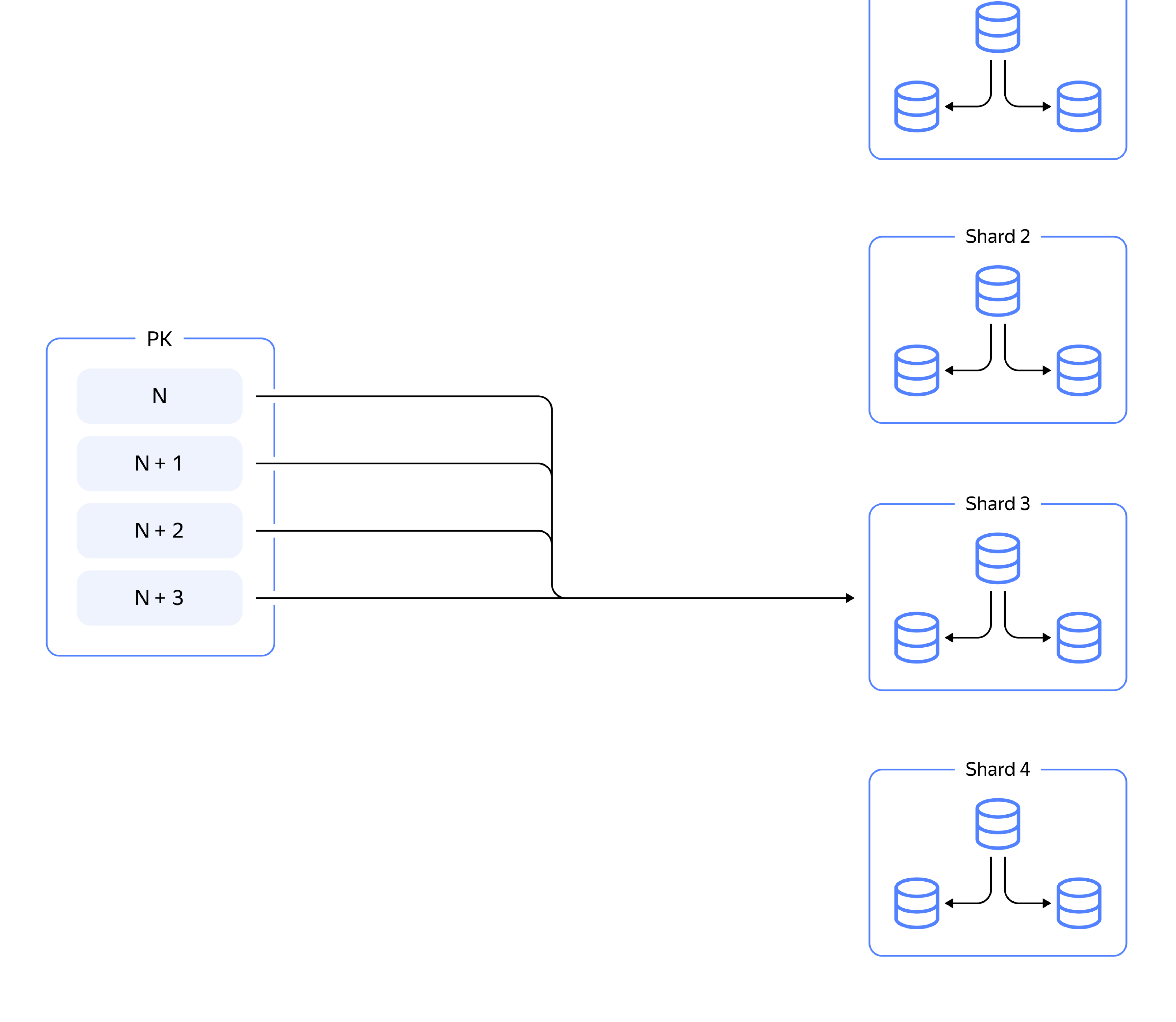


1. Дисклеймер
2. Что мы хотим от шардирования?
3. Что мы пробовали?
4. Stateless Postgres Query Router
5. Демо
6. Открытый процесс разработки

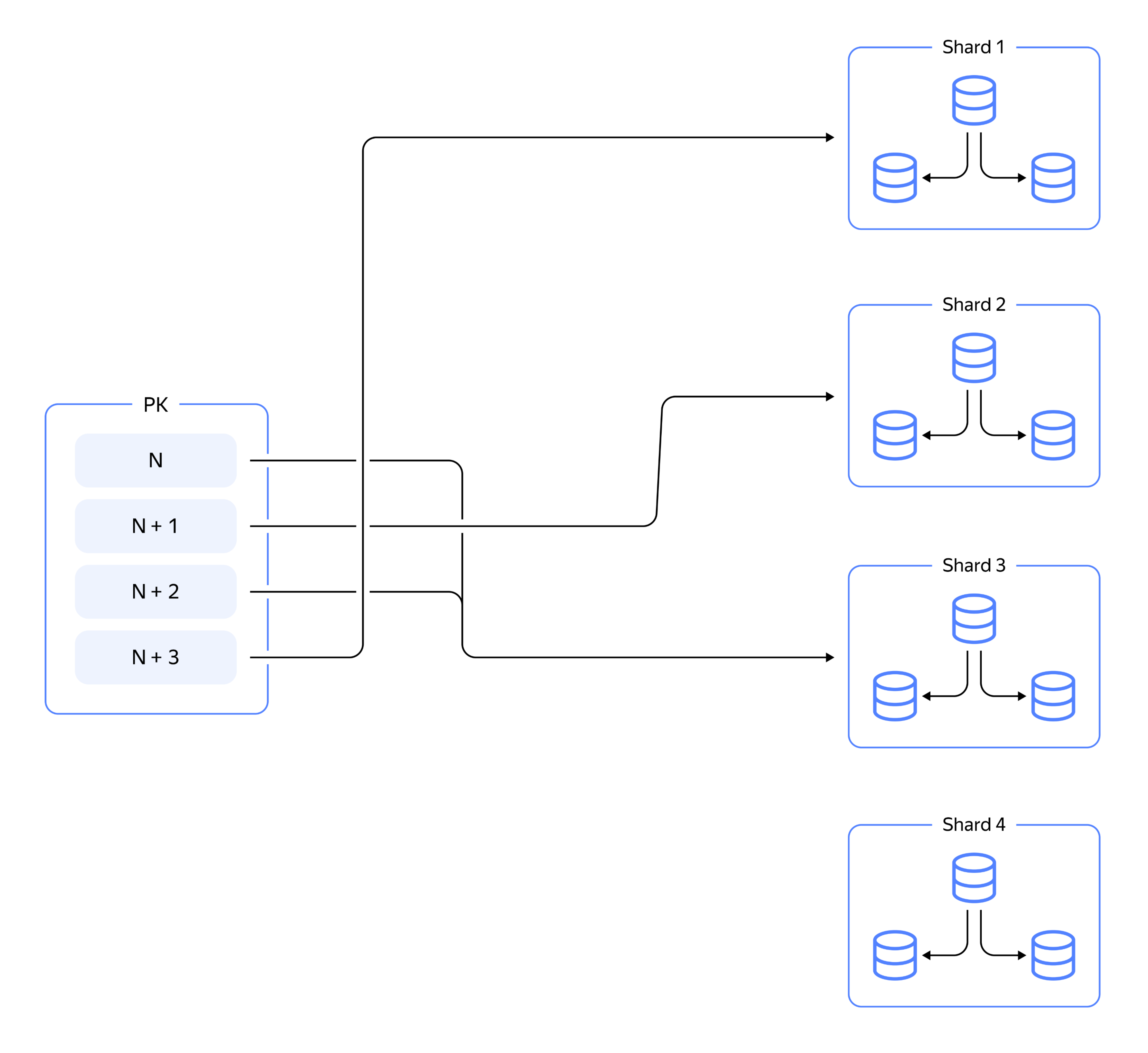
Router

```
postgres=# INSERT INTO x (id)VALUES(14);
NOTICE: send query to shard(s) : shard02
INSERT 0 1
postgres=#
```

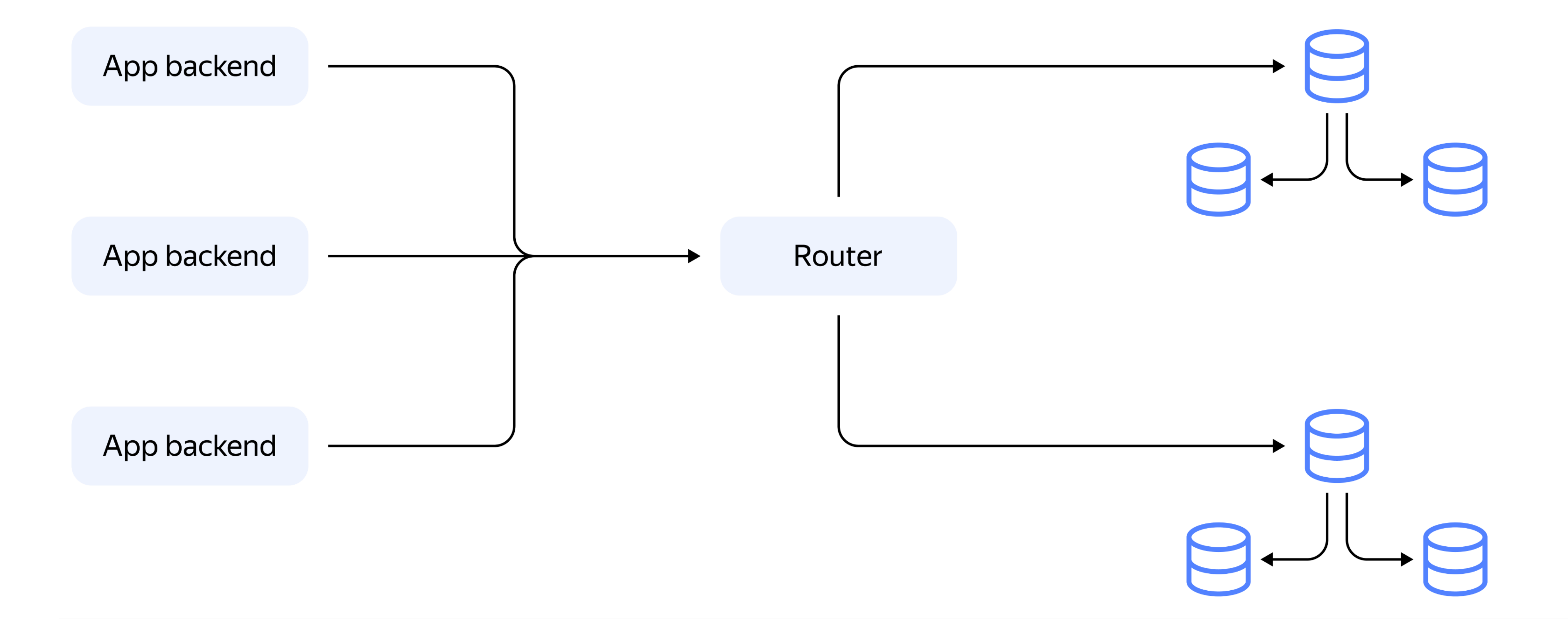
Hash or not to hash?



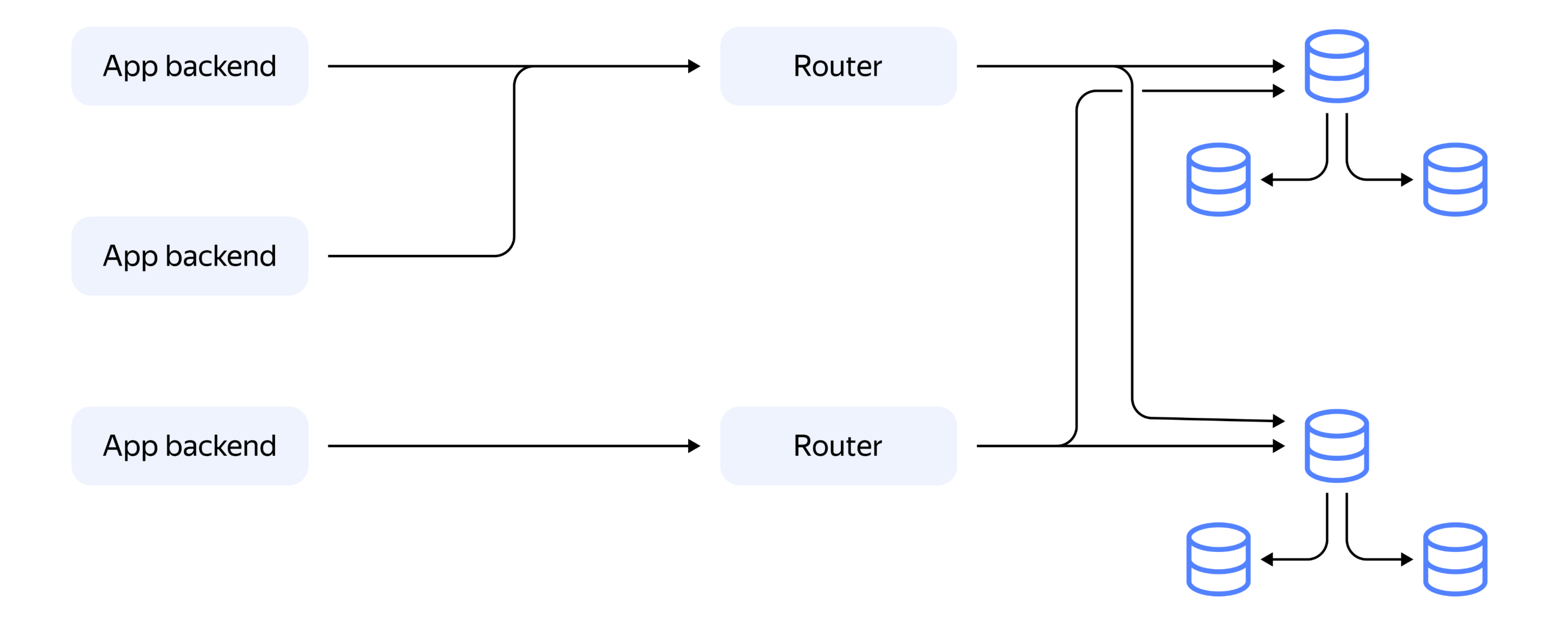
Hash or not to hash?



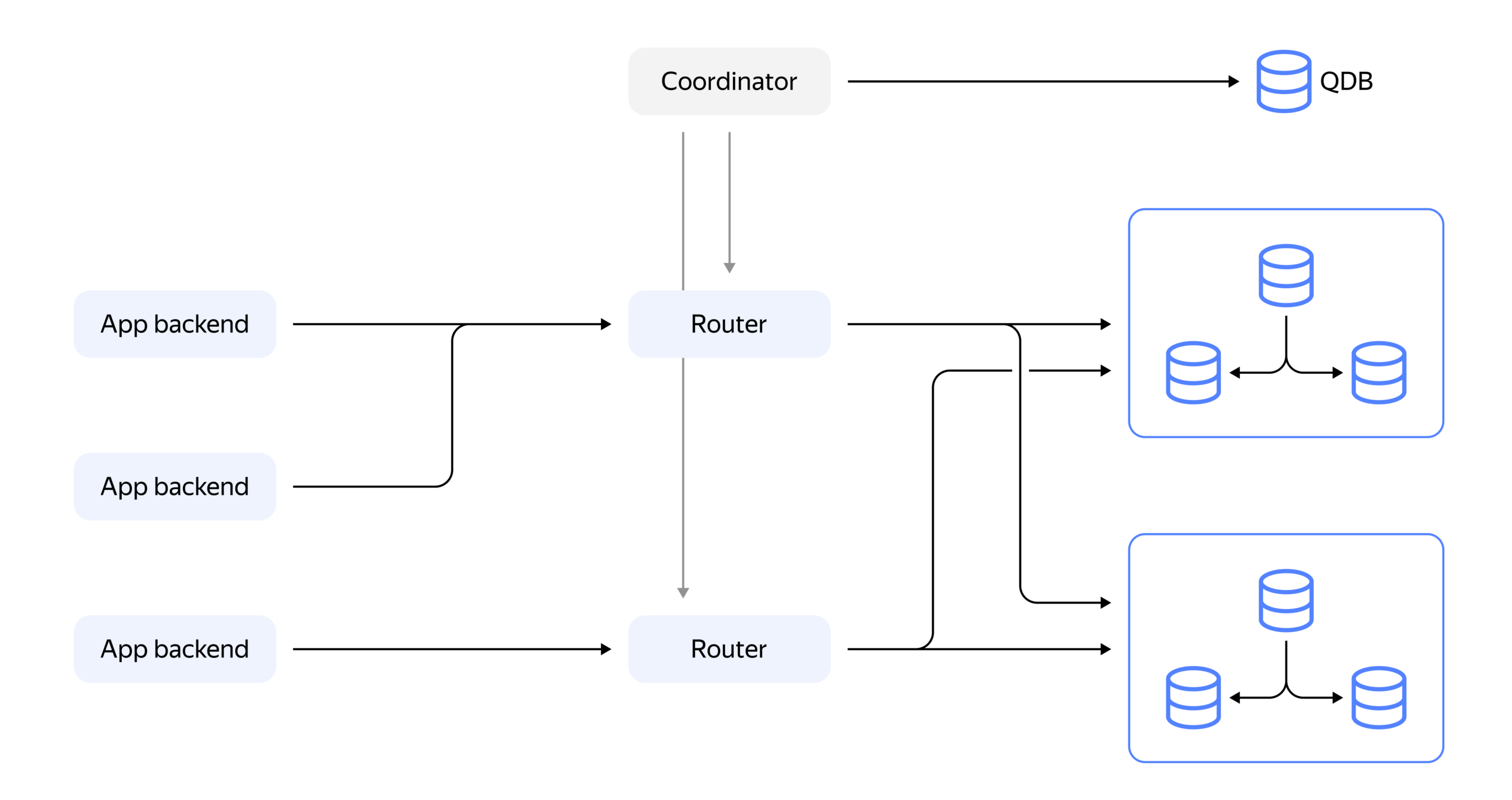
Many routers



Many routers



Coordinator

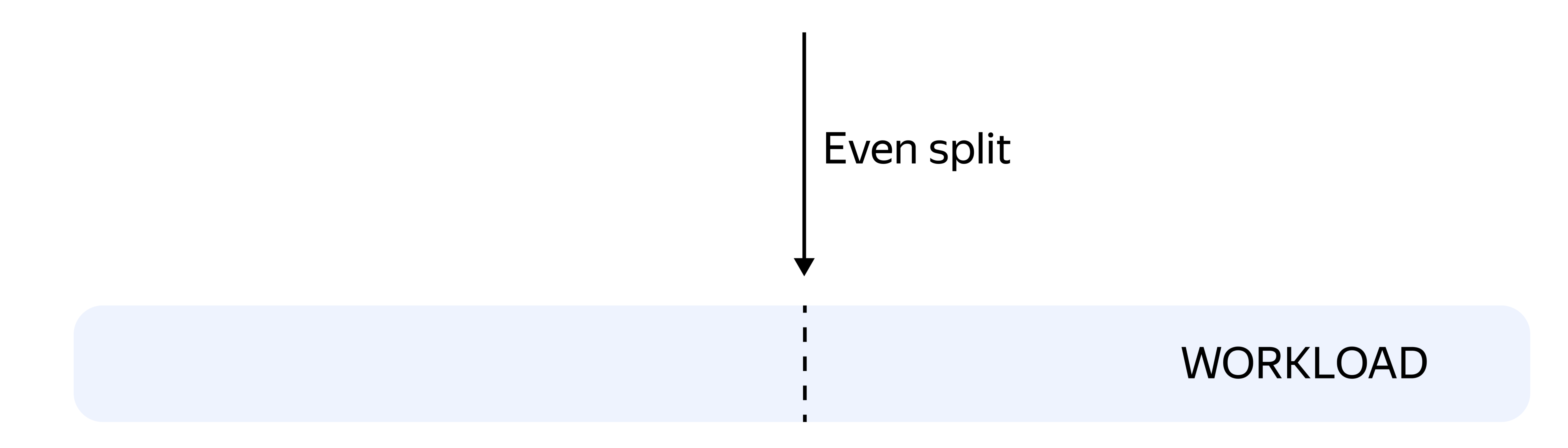


Coordinator

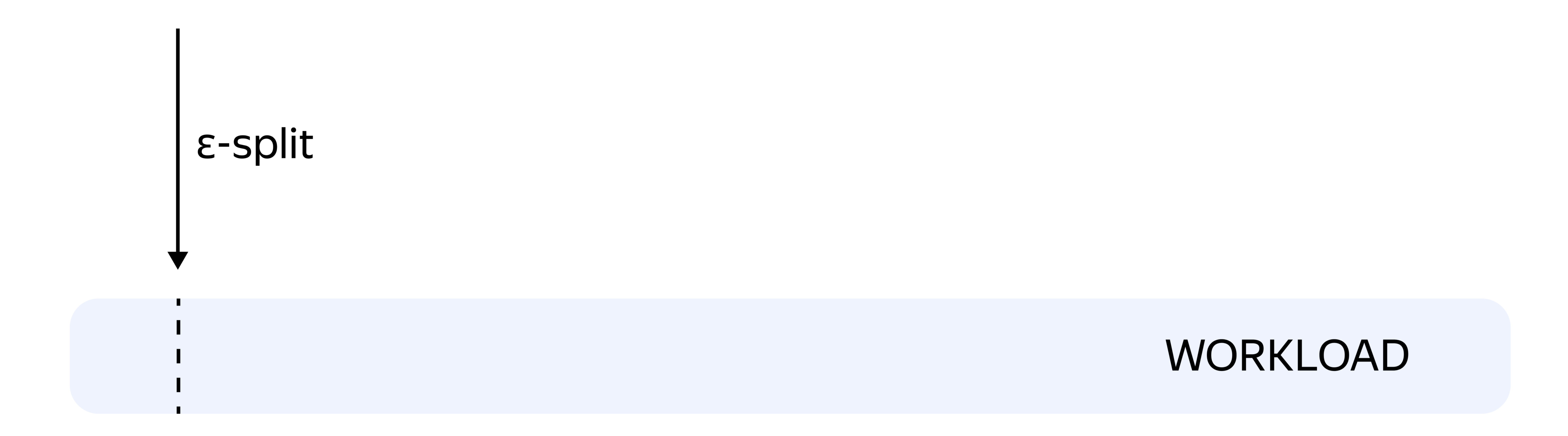
- Shard rebalancing
- Managing routers
- Managing Key Ranges
- Recovery (?)

Coordinator

Shard rebalancer

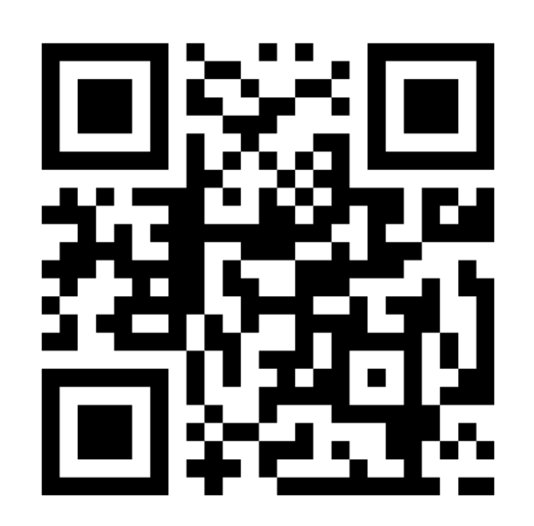


Shard rebalancer



Shard rebalancer

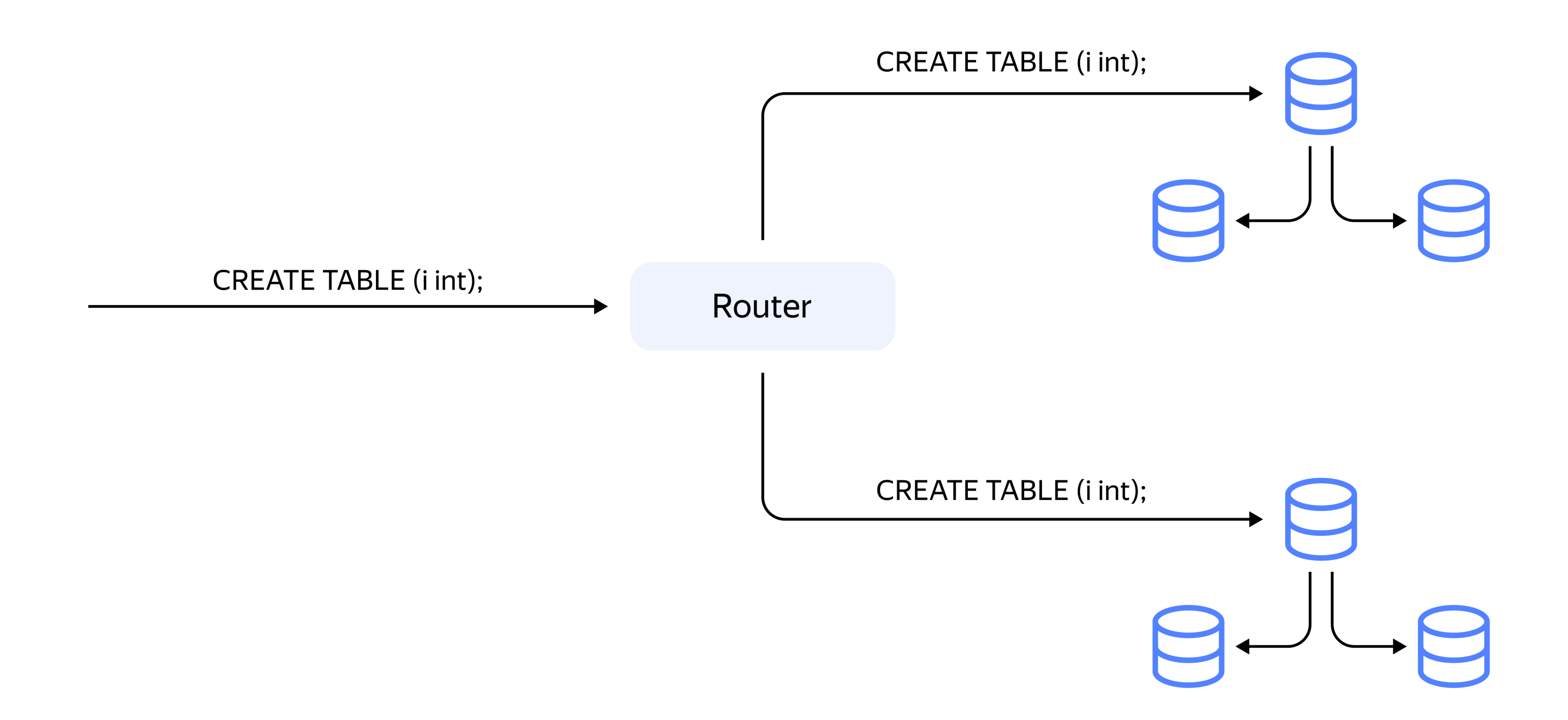
```
> /* a: 1 c: hmm*/ select 1;
> select comment_keys, query_count, user_time from pgcs_get_stats() limit 1;
-[ RECORD 1 ]+----------------------
comment_keys | {"a": "1"}
query_count  | 1
user_time    | 6.000000000000363e-06
```



clck.ru/32XeY5

Multishard Queries

User is allowed to do DDL using Router Connection, which will multiplex query to shards



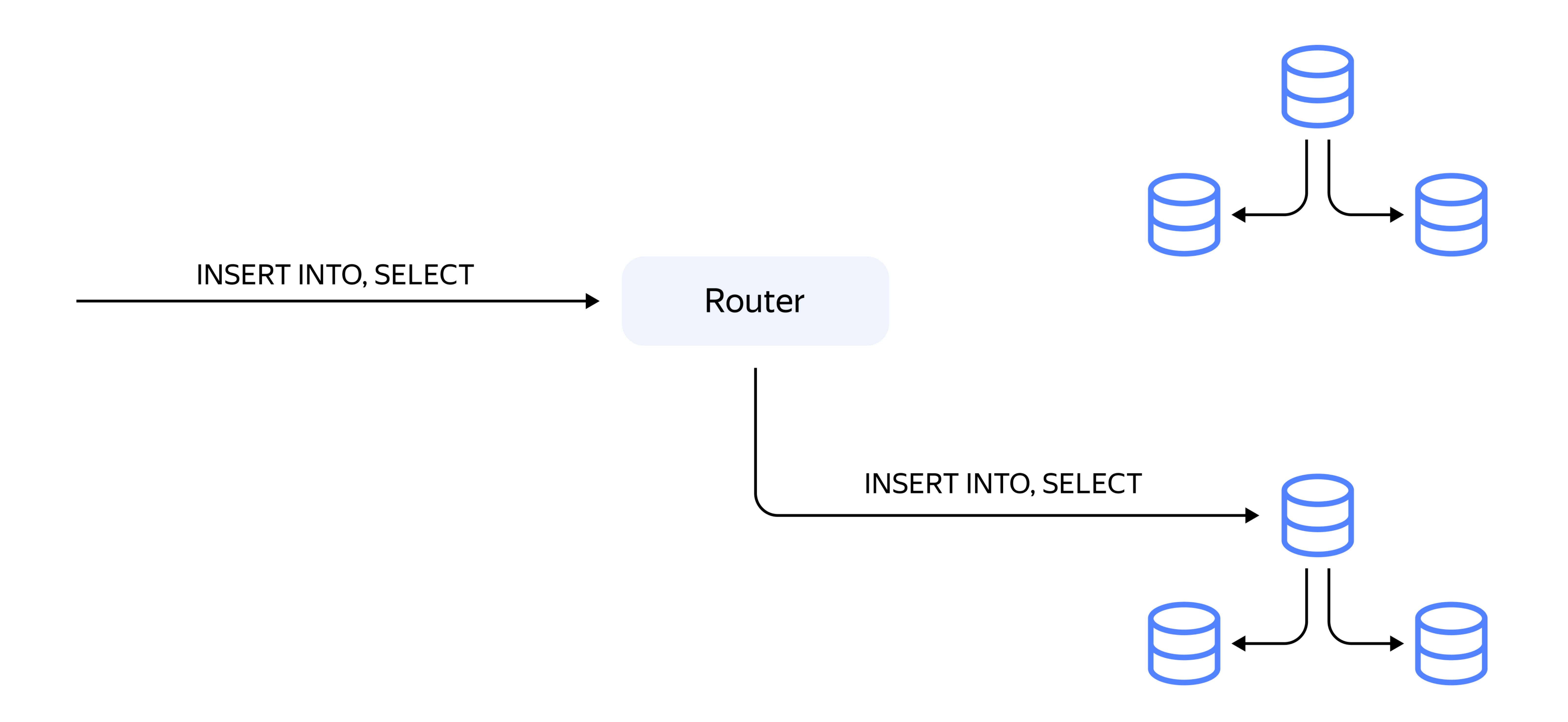
Multishard Queries

```
news=> CREATE TABLE articles (
   id SERIAL NOT NULL PRIMARY KEY,
   url TEXT NOT NULL UNIQUE,
);
NOTICE: send query to shard(s) : shard01,shard02
CREATE TABLE
```

Multishard Queries

```
news=> SELECT * FROM articles;
NOTICE: send query to shard(s) : shard01,shard02
    id     |                  url                 |
-----------+--------------------------------------+
      1235 | https://www.nature.com/articles/d41586 |
 1073741825 | https://news.ycombinator.com/item?id=x |
(2 rows)
```

Single Shard Queries



Single Shard Queries

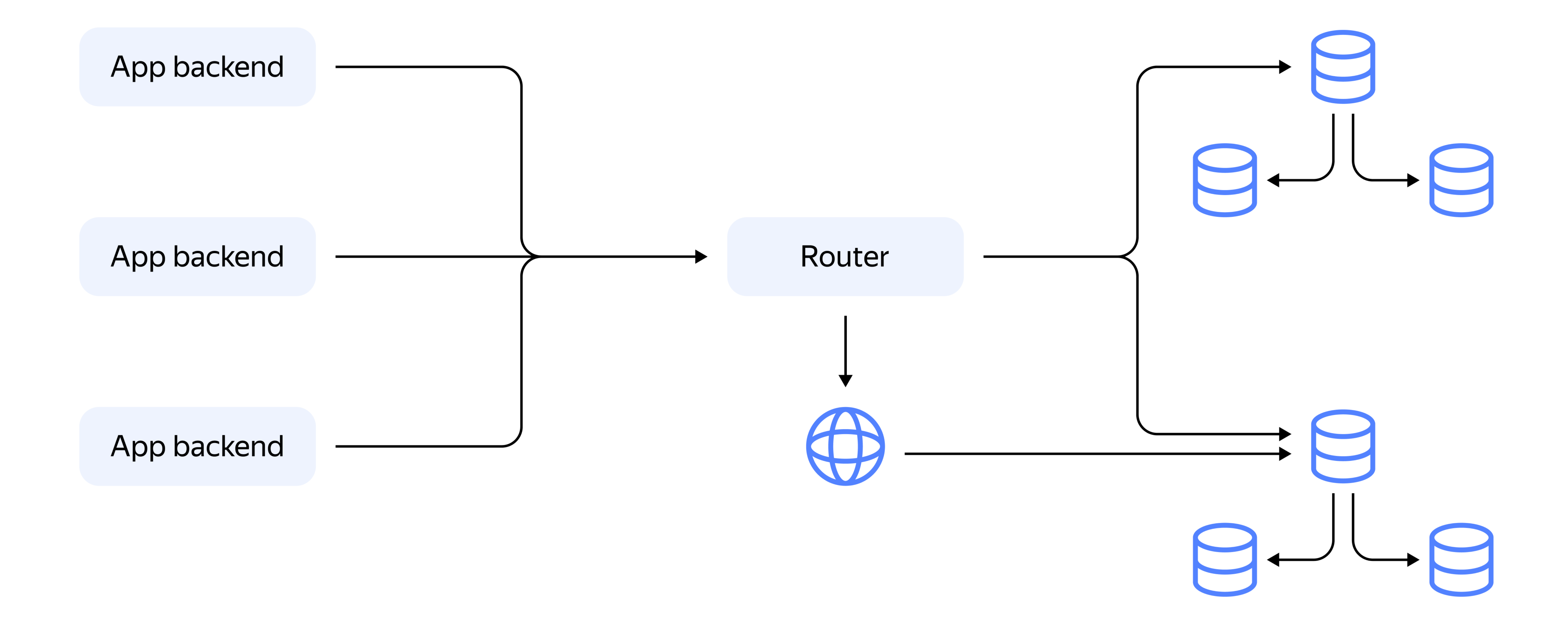
```
news=> INSERT INTO articles (id, url) VALUES (...);
NOTICE: send query to shard(s) : shard01
INSERT 0 1
```

Single Shard Queries

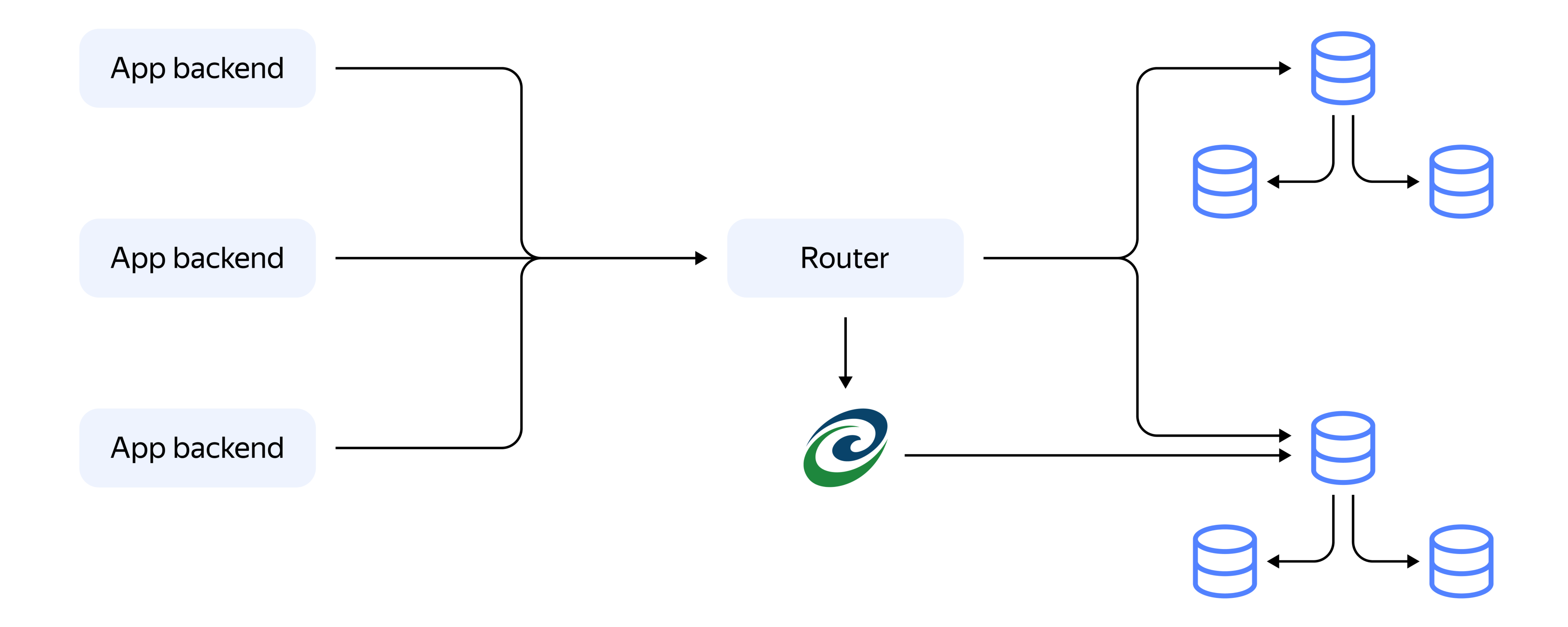
```
news=> SELECT * FROM articles WHERE id > 123;
NOTICE: send query to shard(s) : shard01
  id  |                  url                  |
------+---------------------------------------+
 1235 | https://www.nature.com/articles/d41586 |
(1 row)
```

NB: Sharding key may be composite

World Shard



Citus compatibility?



Latency costs

Latency average (ms)

tps	Postgres	Pgbouncer	SPQR log	SPQR	PgCat
500	0,910	1,115	1,266	1,016	1,493
1000	0,813	1,034	4,236	1,178	1,512
1500	0,785	1,015	—	1,261	1,517
2000	0,788	0,990	7,656	1,282	1,573
2500	0,771	0,971	12,289	1,347	1,664
2700	0,770	0,972	52,701	1,314	1,797
3000	0,767	0,979	1249,946	1,418	1,748
5000	0,761	0,846	—	1,537	2,265

* Postgres: Docker container (Version 13)

** Pgbouncer: balancer to Docker container

gitlab.com/postgres-ai/postgresql-consulting/tests-and-benchmarks/-/issues/30

Нужны пруфы, Билли

1. Дисклеймер
2. Что мы хотим от шардирования?
3. Что мы пробовали?
4. Stateless Postgres Query Router
5. Демо
6. Открытый процесс разработки

github.com/pg-sharding/news-demo

Собирает статьи из разных источников и состоит из:

- RSS parser
- API Server

```
CREATE TABLE articles (
    id SERIAL NOT NULL PRIMARY KEY,
    url TEXT NOT NULL UNIQUE,
    title TEXT NOT NULL,
    description TEXT NOT NULL
);
```

Запросы

Использует внутри только два запроса

```
-- Read articles
SELECT * FROM articles;


-- Create an article
INSERT INTO articles (id, url, title, description)
  VALUES (murmurHash(a.URL), a.URL, a.Title, a.Description);
```

Как запустить

```
export DATABASE_URL="postgres://news@localhost:5432/news?.."

make

./rssparser

./apiserver
```

Как шардировать

1

Собрать SPQR
из исходников

2

Запустить
SPQR-router

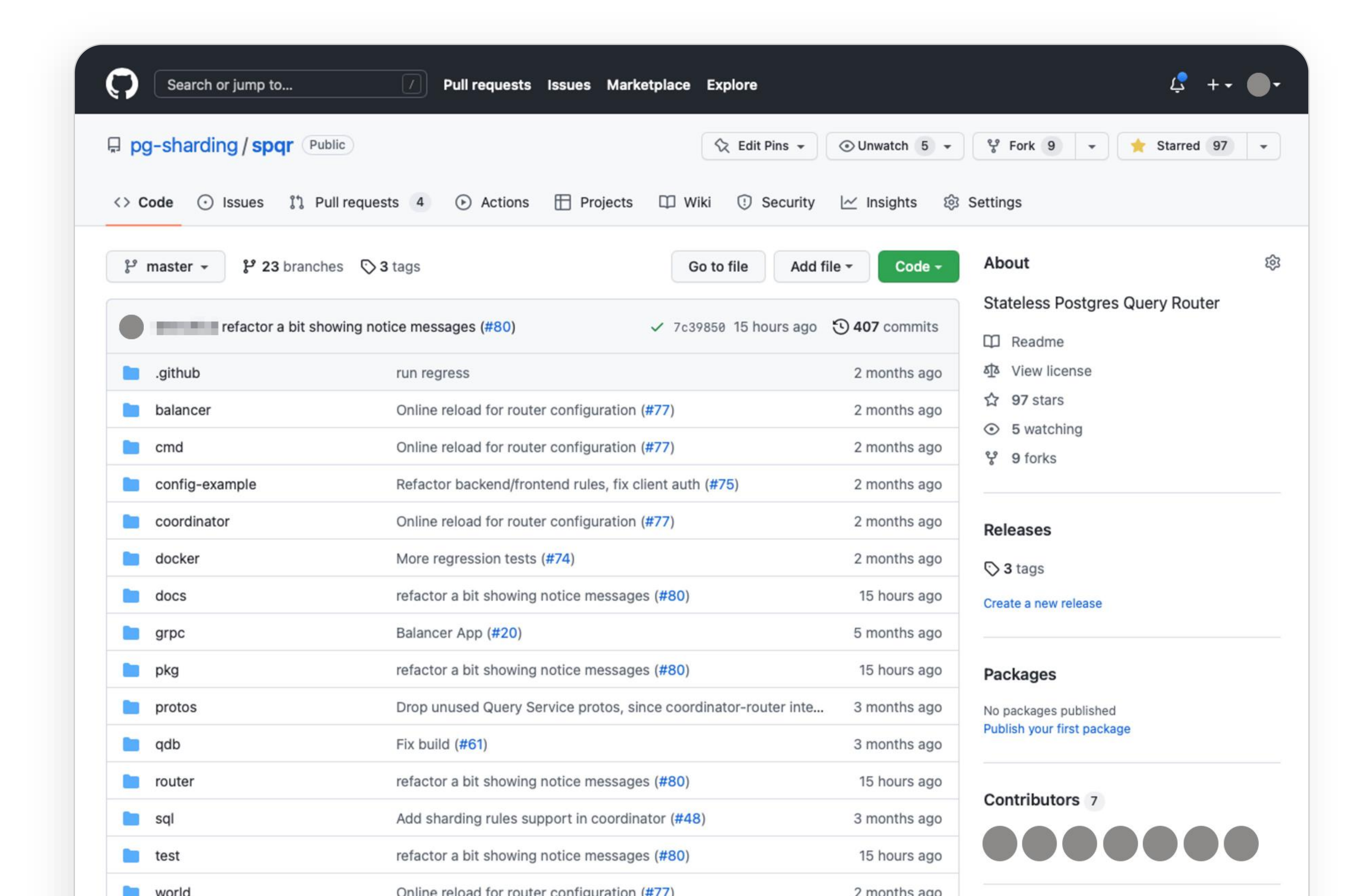
3

Настроить
шардирование
в Router Admin
Console

4

Запустить
приложение

1. Собрать из исходников



2. Запустить

Вызвать ./spqr-rr run -c config.yaml

```
// some common settings
rules:
  frontend_rules:
   - // User, DB, Auth, Pooling
  backend_rules:
   - // User, DB, Auth, Pooling
  shards:
    shard01:
      // Connection settings
    shard02:
      // Connection settings
```

3. Административная консоль

```
→ news-demo git:(main) ✗ psql "host=localhost sslmode=disable
user=news dbname=news port=7432„

            SQPR router admin console
      Here you can configure your routing rules
------------------------------------------------
      You can find documentation here
https://github.com/pg-sharding/spqr/tree/master/docs
psql (14.5 (Homebrew), server console)
Type "help" for help.
```

3. Административная консоль

```
news=> SHOW shards;
    listing data shards
---------------------------
 datashard with ID shard01
 datashard with ID shard02
(2 rows)
```

3. Административная консоль

```
news=> ADD SHARDING RULE rule1 COLUMNS id;
      add sharding rule
------------------------------
 created sharding column [id]
(1 row)
news=> ADD KEY RANGE krid1 FROM 1 TO 1073741823  ROUTE TO shard01;
             add key range
----------------------------------------
 created key range from 1 to 1073741823
(1 row)
news=> ADD KEY RANGE krid2 FROM 1073741824 TO 2147483647  ROUTE TO shard02;
```

3. Административная консоль

```
news=> SHOW sharding_rules;
        listing sharding rules
--------------------------------------------
 sharding rule rule1 with column set: [id]
(1 row)
news=> SHOW key_ranges;
 Key range ID | Shard ID | Lower bound | Upper bound
--------------+----------+-------------+-------------
 krid1        | shard01  | 1           | 1073741823
 krid2        | shard02  | 1073741824  | 2147483647
(2 rows)
```

4. Подключиться к роутеру

```
→ news-demo git:(main) ✗ psql "host=localhost port=6432"
psql (13.3, server 9.6.22)
Type "help" for help.
dbname=> CREATE TABLE articles (
dbname(>     id SERIAL NOT NULL PRIMARY KEY,
dbname(>     url TEXT NOT NULL UNIQUE,
dbname(>     title TEXT NOT NULL,
dbname(>     description TEXT NOT NULL
dbname(> );
NOTICE: send query to shard(s) : shard01,shard02
CREATE TABLE
```

4. Подключиться к роутеру

```
dbname=> INSERT INTO articles (id, url, title, description)
VALUES ('1235', 'https://www.nature.com/articles/d41586-022-01516-2‘,
'Science needs more research software engineers', 'nope');
INSERT 0 1
dbname=> INSERT INTO articles (id, url, title, description)
VALUES ('2147483644', 'https://www.nature.com/articles/d41586-022-01516-2‘,
'Science needs more research software engineers', 'nope‘);
INSERT 0 1
```

4. Подключиться к роутеру



```
news=> SELECT * FROM articles WHERE id > 1;
NOTICE: send query to shard(s) : shard01
 id  |                    url                    |                  title                  | description
------+-----------------------------------------------------+-----------------------------------------------+-------------
 1235 | https://www.nature.com/articles/d41586-022-01516-2 | Science needs more research software engineers | nope
(1 row)
news=> SELECT * FROM articles WHERE id < 2147483647;
NOTICE: send query to shard(s) : shard02
    id     |                  url                  |                       title                       | description
------------+------------------------------------------------+------------------------------------------------------------+-------------
 1073741825 | https://news.ycombinator.com/item?id=29201000 | Scalable PostgreSQL Connection Pooler (github.com/yandex) | nope
(1 row)
```

1. Дисклеймер
2. Что мы хотим от шардирования?
3. Что мы пробовали?
4. Stateless Postgres Query Router
5. Демо
6. Открытый процесс разработки

Как помочь?

```
git clone https://github.com/pg-sharding/spqr
cd spqr
make run
```



Задавай вопросы
в Telegram-чате

clck.ru/32Y3TS



Отправляй
pull requests
и issues

clck.ru/32Y3Vw



Запусти демоприложение

clck.ru/32Y3Zr

Спасибо!

Денис Волков

Open Source RDBMS
Development Team Yandex
Cloud

denchick@yandex-team.ru

Кирилл Решке

Open Source RDBMS
Development Team Yandex
Cloud

reshke@yandex-team.ru



Яндекс



Обратная связь
и комментарии
к докладу по ссылке





HighLoad++ 2022

Яндекс